

ICS HW ans 2

T1

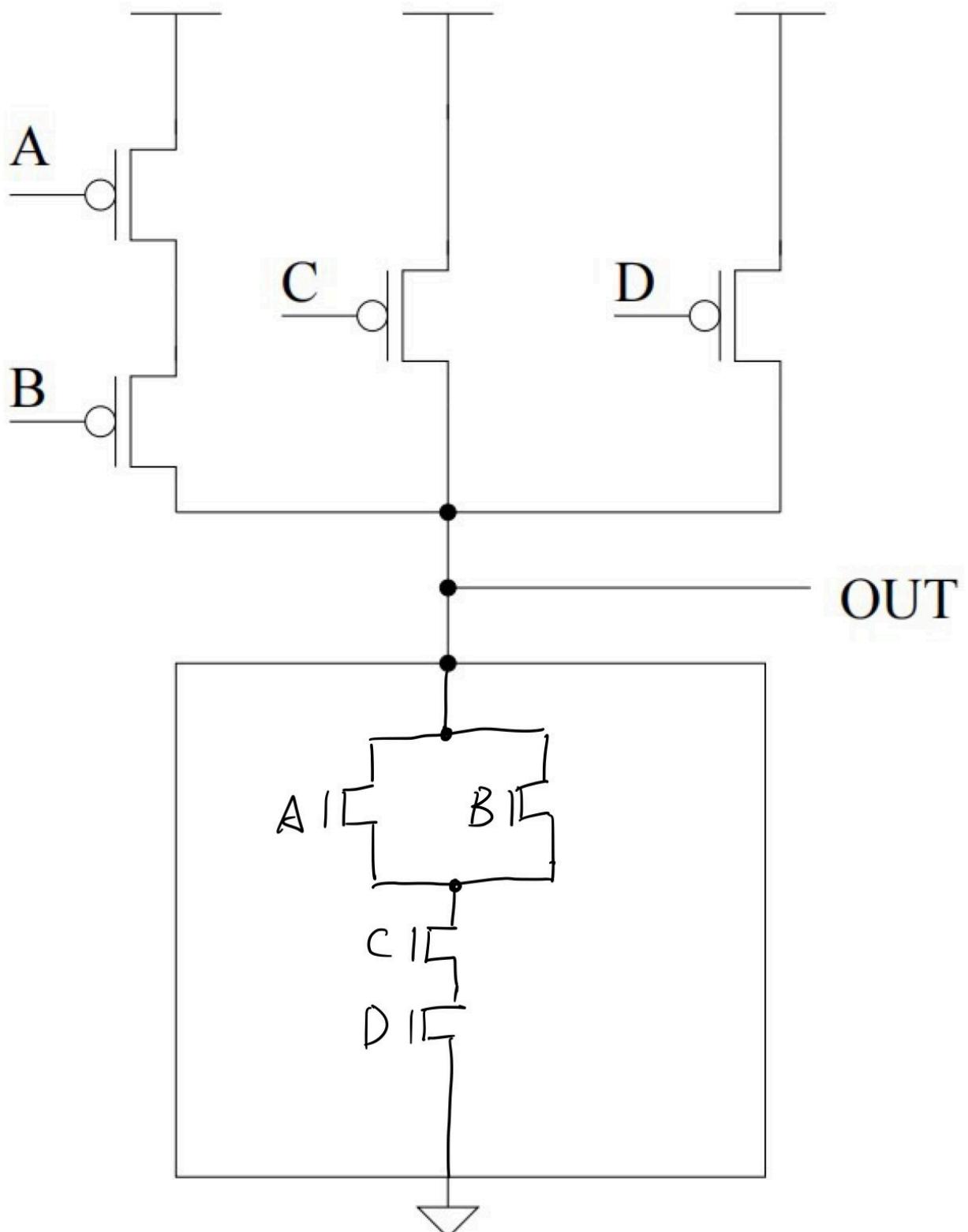
- 0 OR X = **X**
- 1 OR X = **1**
- 0 AND X = **0**
- 1 AND X = **X**
- **0** XOR X = X

T2

1.

A	B	C	Out
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

2.



T3

1.

$$\text{NOT } A = A \text{ NOR } A$$

$$A \text{ OR } B = (A \text{ NOR } B) \text{ NOR } (A \text{ NOR } B)$$

$$A \text{ AND } B = (A \text{ NOR } A) \text{ NOR } (B \text{ NOR } B)$$

2. 如果异或门可以构成完全集，那么可以实现与逻辑。按照与逻辑的真值表：

X	Y	Output
0	0	0
0	1	0
1	0	0
1	1	1

$X \oplus Y = X + Y \pmod{2}$, 仅用异或逻辑可以实现的所有逻辑函数可以表示成若干个X,Y,1,0的异或运算的累加, 即

$$Output = m_1 * X + m_2 * Y + m_3 * 1 + m_4 * 0$$

带入X和Y可以得到:

$$\begin{cases} m_3 \equiv 0, \\ m_2 + m_3 \equiv 0, \\ m_1 + m_3 \equiv 0, \quad m_1 + m_2 + m_3 \equiv 1 \end{cases} \pmod{2}$$

矛盾。所以异或门不能单独构成完全集。

T4

$$1. 0\ 00000001\ 0000000\ 00000000\ 00000000(2) = 1 * 2^{-126}$$

$$2. 0\ 00000000\ 1111111\ 11111111\ 11111111(2) = 0.1111111111111111111111111 * 2^{-127} = 2^{-126} - 2^{-126-23} = 2^{-126} - 2^{-149}$$

$$0\ 00000000\ 0000000\ 00000000\ 00000001(2) = 2^{-149}$$

非规格正数的取值范围为 $[2^{-149}, 2^{-126} - 2^{-149}]$

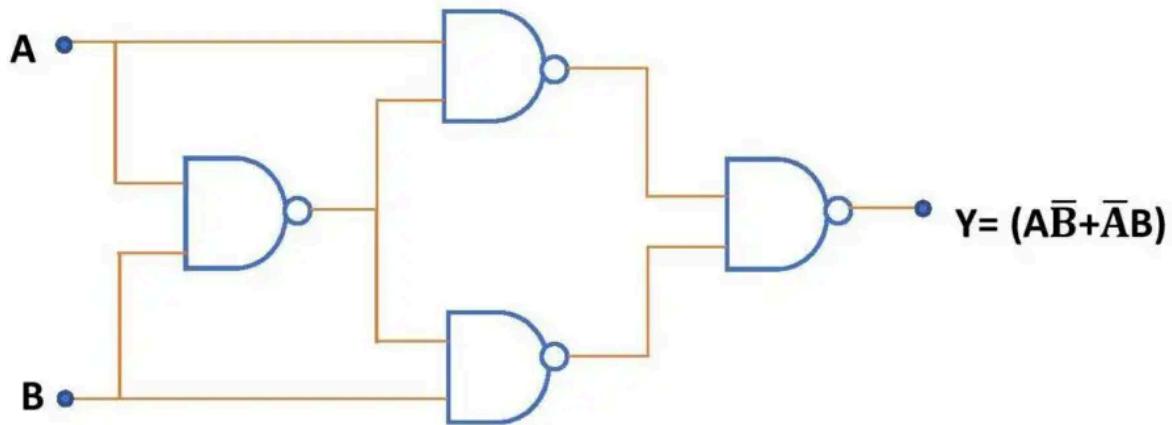
3. 如果没有非规格化数, 当计算结果小于最小规格化数 2^{-127} 时将变得无法表示, 而且原有系统不存在0的表示方法。

4. 浮点数的分布不是均匀的。

对于同一个指数E, 尾数M的变化使得数值在区间 $[2^E, 2^{E+1}]$ 内是均匀分布的, 但相邻数值的绝对间隔是 2^{E-23} 。

在非规格化数区间, 数值呈均匀分布, 绝对间隔恒定为 2^{-149} 。

T5



T6

1. 为了可以写入第一个位置， $A[1:0]$ 应当为 00，同时为了写入 WE 应当为 1
2. 寻址空间：所能表达的最大的不同地址。寻址能力：每个地址位置上存储的数据位数。本图中寻址空间为 4，寻址能力为 4
3. 为了将寻址能力提升到 k bit，则需要将每个位置存储的 bits 提升为 k 位，对应则是在每个位置添加 $k-4$ 个门控 D 锁存器。

T7

48位的操作码至少需要6位进行编码，28个寄存器至少需要5位进行编码，所以 $32-6-5-5=16$

16位的二进制补码所能表示的范围为 -2^{15} 到 $2^{15} - 1$ 。IMM 字段可以表示的数值范围是 **-32768 到 32767** (十进制)。

T8

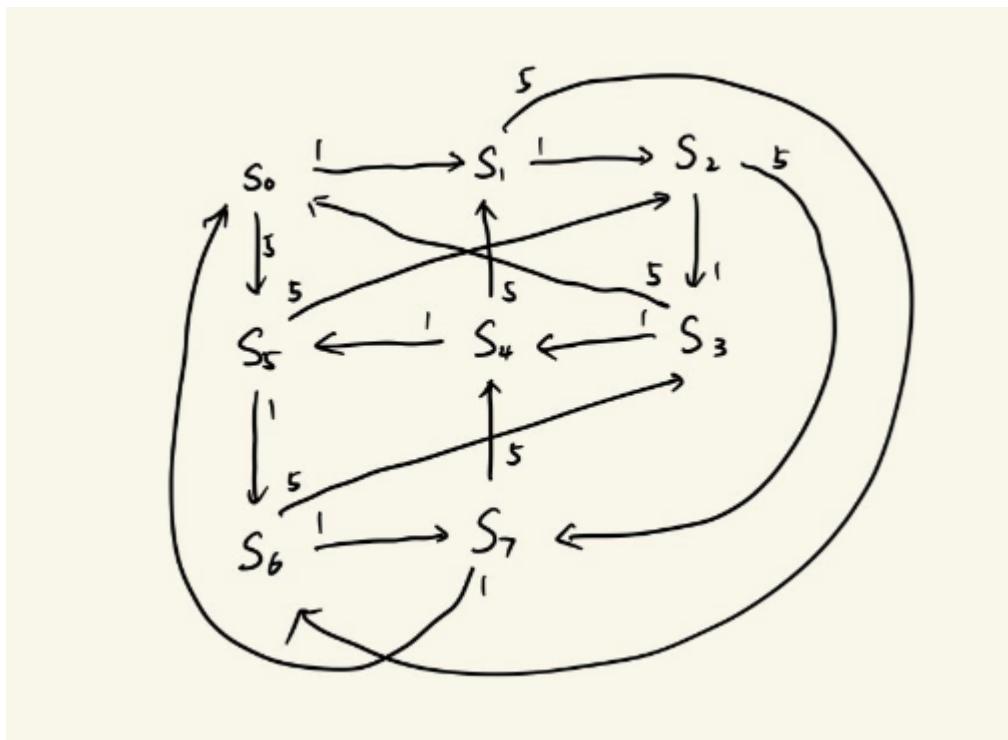
1. Memory, Processing Unit, Control Unit, Input and Output
2. 程序的核心思想：将程序指令和数据以二进制形式共同存储在同一个读写存储器（内存）中。由一个中央处理单元（CPU）顺序地从内存中读取指令并执行。一个称为程序计数器（PC）的寄存器负责指向一条要执行的指令地址，通常通过自动递增来实现顺序执行。
(言之有理即可) 量子计算机不能使用冯·诺依曼架构，是因为量子信息的存储、传输和处理方式与经典信息根本不同，特别是在“可复制性”“测量”“逻辑运算”等方面存在物理层面的限制。
 1. 量子态无法被自由读取或复制
 2. 量子逻辑运算是幺正操作
 3. 量子存储与处理无法物理分离
 4. 量子测量破坏程序状态

T9

1. 110000
2. 12
3. 6
4. 6

T10

饮料机中剩余金额的有限状态机转化图如下所示，状态的下标代表了剩余金钱。



T11(言之有理即可)

1. 可以将每瓶酒进行编码，比如第一瓶为0001，第二瓶为0010以此类推，第八瓶为1000。其中每个酒瓶的编号中的1代表了这瓶酒要给对应的小鼠喝。等到第二天我们就可以根据死掉小鼠的编号推断出哪瓶酒有毒。
2. 最多可以检测 $2^4 - 1$ 瓶酒，如果从0开始编码的话可以检测 2^4 瓶酒
- 3.

M1	M2	M3	M4	死亡总数
0	0	0	1	1
0	0	1	0	1
0	0	1	1	2
0	1	0	0	1
0	1	0	1	2

M1	M2	M3	M4	死亡总数
0	1	1	0	2
0	1	1	1	3
1	0	0	0	1

$$\frac{4}{8} * 1 + \frac{3}{8} * 3 + \frac{1}{8} * 3 = \frac{16}{8} = 2$$

如果从0开始编码的话可以减少一种表示方法，我们不妨将表示7的那一瓶酒用0000表示，这样的话预期死掉 $\frac{13}{8}$ 只老鼠